More on Internal Ptrs

Sign bit restriction

Mon, Oct 21, 2002

Until recently, the size of an internal ptr has always been 4 bytes. An internal ptr is the intermediate representation of an ident that used by a read-type routine, along with the requested number of bytes, to produce reply data in response to a data request, whether it be in the Classic protocol or the Acnet `RETDAT` protocol. Now there has been implemented the option to increase the size of an internal ptr structure beyond 4 bytes for a particular listype in which the ptr-type number is not such as to refer to a system table number (< 32). A ptr-type routine, along with its related read-type routine, can be designed to use any number of longwords for an internal ptr structure. It is thought that this will first be useful in generic support for moderately fast digitizers, including 1KHz and 10 KHz versions.

For server requests, an internal ptr format always has the same form. The sign bit is set to indicate that the remaining 31 bits make up a memory address that refers to a buffer in which the data sought already resides, such as an external answer buffer that is part of the request block, or associated answer block, that supports the request. It is used for data that comes from other nodes, and when such replies are received, they are stored in such a buffer. When the request is initialized, the internal ptr is set to an address that points to this buffer, with the sign bit set to indicate this case. Without the sign bit set, the read-type routine must come up with the data itself, as it has not previously been worked out by another node and cached into the buffer.

Some people have suggested that it is not appropriate to signal the server case by usurping the sign bit of the 4-byte internal ptr. Rather, some other means should be followed to recognize this case, so the internal ptr can gain back the use of the sign bit. In the default vxWorks memory address allocation for the MVME-2401, the PMC space would have addresses that are above `0x80000000`, and such addresses might easily be ones to which an internal ptr should refer.

One solution for this requires an additional array that can be parallel to the internal ptrs array. It could be a bit array or a byte array, and it could be tested for each corresponding internal ptr to determine whether it is merely an address where the data can be found all ready to deliver, or something else that must be interpreted to help come up with the reply data. In order to make use of it, this additional array would have to be made available to each read-type routine. As a result, it would require modification of the code of every read-type routine. But it is possible, and it would free up the sign bit in internal ptrs, allowing such cache buffers to be located anywhere in memory space.